Multiple Signature Handling in Workflow Systems *

Karl R.P.H. Leung  
Department of Computing and Mathematics  
Hong Kong Institute of Vocational Education (Tsing Yi)  
Hong Kong  
kleung@computer.org

Lucas C.K. Hui  
Department of Computer Science and Information Systems  
The University of Hong Kong  
Hong Kong  
hui@csis.hku.hk

Abstract

In paper-based workflow systems, signatures by individual or groups of people have been used extensively for various purposes. Currently there are many studies on computerizing workflow systems. Also there are studies on implementing signatures in electronic media. But the diversified purposes of a signature in workflow makes a straightforward implementation of digital signature schemes inadequate to satisfy the needs of an electronic workflow system. There are few studies on the implication on the purposes of signatures in workflow systems after the change in the media of operation. Signatures in workflows can be classified into two categories namely, single and multiple signatures. This paper reports our studies on the purposes of multiple signature and our design for the implementation of these purposes with illustration using the Liaison workflow engine.

1 Introduction

Signatures are widely used in paper-based workflow systems. Different purposes of using signatures include authorization, accountability, witness, and time stamp[7, 11]. Various techniques of handling signatures are based on the physical properties of paper and human interpretation. For example, carbon copies are used to distribute a handwritten signature to more than one recipient. These techniques are mature and are widely used in daily paper-based processing.

The development of information technology leads to the migration of workflow systems from paper-based to electronic-based media. Some of the purposes of signatures in paper-based workflow processing may no longer be necessary. For example, a computer system can time-stamp a document automatically by putting a digital signature over the document, giving the processing time and date. Digital signature schemes are well studied and those schemes are usually based on cryptography such as the public key crypto-systems [15]. Emphasis is placed on the integrity of the signed data, as well as the non-repudiation properties of the digital signature. However, the purposes of signatures in workflow are usually ignored in the study of digital signatures. In an electronic workflow system, different procedures are required to handle digital signatures for different purposes. For example, signatures for authorization purposes have to be checked against the organization structures. This is because the organization structure can record the information about which officer has the right to authorize the usage of certain resources. Handling the signatures without referencing the organization structure may lead to inappropriate handling of the workflow process. Therefore there is a need to integrate the digital signature processing scheme and the enterprise workflow model together[7].

Signature handling can be classified into two categories namely, single signature and multiple signature. Single signature handling refers to the cases where only one person will sign a document, while multiple signature refers to the cases where more than one persons will sign a single document. Note that in contemporary commercial environment, many decision making processes require the signature of more than one person. Therefore multiple signatures are very important. In [11], we reported our studies on single signatures in workflow and our implementation of these purposes in the Liaison Workflow Engine [10] which is the architectural design of Liaison Workflow Model [2, 4]. In this paper, we are going to report our study on multiple signatures and a generic design for implementation. These are illustrated using the Liaison Workflow Engine Architecture.

Section 2 discusses some background of workflow sys-
2 Background

Most of the research on workflow such as METEOR [8] and WfMC [9], focuses on the management of workflow with different kinds of technology such as database. Few articles include signature in their studies. Signatures, at best, is only a kind of operations in the tasks. In these research systems, like many commercial workflow products (such as FlowMark [12]), encrypting a form is almost the base level of abstraction [15]. Hence the purposes of the signatures inside a form are outside the scope of their systems.

Asymmetric Cryptosystem [14, 5] is the fundamental tool for providing digital signature services to workflow management systems. This is a tool for signing and validation according to the key. The purposes of the signatures and the information associated with the signature, such as the validity of the actor who signs the document, is not included in this basic mechanism. With cryptography features, workflow users have to investigate how to apply these features to satisfy the signature requirements on their own.

Furthermore, in addition to the data to be signed, different purposes of signature require other information such as knowledge of the organisation structure. Comprehensive organisation structure, processes, data, and corresponding relations between these aspects are then required to be captured in workflow modelling in order to support different signature purposes. Most workflow models and commercial products do not capture sufficient information to support different signature purposes.

2.1 Liaison Workflow Model and Liaison Workflow Engine Architecture

The Liaison workflow model [2, 4] is an extension of the reference workflow model proposed by the Workflow Management Coalition [9]. The design of Liaison is based on the requirements of workflow systems reported in [1]. In addition to the model, it is also equipped with a workflow specification language Valmont [3]. Like the reference workflow model, Valmont captures the fundamental elements of the workflow paradigm: Organisation Model, Information Model, Process Model, and their relationships. Unlike the reference workflow model, it supports a rich organisation model and sophisticated activity assignment constraints.

The Liaison Workflow Engine was first proposed in [10]. It was extended with the capability of handling different single signature purposes as reported in [11]. The Liaison Workflow Engine consists of two major components: the build-time engine and the run-time workflow engine. The build-time engine is responsible for parsing workflow specifications in Valmont to generate the local data. These include workflow definitions, task definitions, organisation structure and actor information, and, form and data definitions. These local data are kept in the repositories of Workflow Definition, Task Definition, Actor Information, Form & Data Store, respectively (see Figure 1).

The workflow engine is responsible for managing and controlling workflow and task executions, actor assignments and information usage. In Liaison workflow engine, these responsibilities are handled by the co-operation of Clock, Scheduler, Task Manager, Actor Manager, Information Manager, Actor Manager, Actor Interface Manager and Signature Manager. This architecture is shown in Figure 1.

![Figure 1. Liaison Workflow Engine Architecture](image-url)
ing signatures, including key management, signature generation and signature validation. In the signing and validating operations, it also possesses the capability of handling different kinds of signature purposes.

2.2 Digital Signature

Different public key cryptographic schemes have been used to implement digital signature and data encryption [15]. For example, the Rivest-Shamir-Adleman (RSA) scheme [14], Digital Signature Algorithm (DSA) scheme [13, 17], the ElGamal scheme [5], and the elliptic curve digital signature algorithm (ECDSA) scheme [18, 16]. When these schemes are used to implement digital signatures, each user owns a key-pair containing a private key and a corresponding public key. The private key is meant to be known only by the owner, and the public key is known to the public. The mathematical properties of the public key system guarantee the following:

- A message, the plaintext, after encrypting with a private key and then decrypting with the corresponding public key, will become the plaintext itself; and
- Given the public key, there is no computationally feasible way to compute the corresponding private key.

Given these two properties, digital signatures can be implemented. A simple example on how a person Amy sends a signed message M to another person Bob is as follows:

1. Amy encrypts M with her private key. The encrypted message is denoted as S. Amy sends M and S to Bob.

2. After Bob receiving M and S, Bob will try to decrypt S using Amy’s public key.

If the decryption successfully generates a message which is identical to M, then Bob can be sure that the message is originated from Amy, since she is the only person who can make use of her private key to generate S. If S is generated by some other people’s private key, Bob will only receive garble after decrypting with Amy’s public key.

Real life digital signature schemes are elaborated on the above simple example, incorporating technologies including block cipher, public key certificates, and sophisticated key distribution and management methods [6].

Unlike public key cryptography, symmetric key cryptography uses the same key for both encryption and decryption. Since authentication based on symmetric key cryptography does not have the non-repudiation properties [15], digital signature schemes are usually based on public key cryptosystems.

3 Multiple Signatures

In the processing of workflow systems, there are many forms which require the signatures of more than one actor. For example, in a group decision making situation, a form requires the approval by signature of a majority of designated managers. It will be difficult to maintain the special properties of the multiple signature if they are broken down into different single signature in signing and validating. Furthermore, information from the organization model is needed to correctly process forms involving multiple signature.

According to the purposes and operations of multiple signatures, we identified some fundamentals cases. More complicated cases can be composed by these fundamental cases.

3.1 Sequential Multiple Signature

Very often the outcome of a decision making process depends on authorization from different officials. Let us illustrate the sequential multiple signature with the following example. The MIS department of a company would like to renovate its computer room so as to meet the contemporary hardware requirements. The department needs approval from the Financial Controller and the Estate Department. The MIS department then generates a form to obtain the approval from these two departments. This is the case of a “signature on signature”. That is, the first signer signs on the content of the form, then the second signer signs on the content of the form and the first signer’s signature, and so on. Processing of the form can only be possible when the last signature is obtained.

According to different company policies, there will be different design of workflows to handle this multiple signature requirements.

Independent Sequential Multiple Signature The company may only require that the renovation be approved by the two departments. The order of approval is immaterial. Then the MIS department can request the approval from the Financial Controller and the Estate Department one by one.

Dependent Sequential Multiple Signature The company may set up the policy in another way. For any renovation project, it has to be approved by the Financial Controller first. Then the approval from the Estate Office will depend on the approval signature of the Financial Controller.

3.2 Parallel Multiple Signature

Processing of a form can be approved by a group of actors concurrently. In this case, if the form can be dupli-
cated, a copy of the form can be transmitted to each of the actors for signature. This is very common in group decision making. Note that for an electronic workflow system, duplication of forms is always possible. The signature of each signer is on the content of the form, but not on the signature of other signers. We call this mechanism of distributing form copies fork.

Also there is a need of a separate mechanism to collect the signatures and thus continue the processing of the form. We call this mechanism join. According to the operations of sign and validate, fork and join can be further classified as follows.

3.2.1 Fork

Depending on the choice of all or some of the actors selected to sign the form, there are two types of forks namely, fork-all and fork-some.

fork-all In the example, if the MIS department requires only the approvals of the Financial Controller and Estate Office, and the approvals have no dependence, the MIS department can make copies of the approval form and send to both departments concurrently. Then all the tasks following the creation of the renovation form will be invoked concurrently.

fork-some In the same example, in addition to the approval of the Financial Controller and Estate Office, the MIS department may also require the approval from at least three out of six members of the Board of Directors (BoD). Then, besides making copies for the Financial Controller and Estate Office, the MIS manager may choose to send the renovation request form to at least three BoD members. But which BoD members will be given copies of the renovation request form for signature depends on the decision of the MIS manager. The number of copies can range between five to eight.

3.2.2 Join

Similar to fork, depending on the choice of collecting all or some of the signatures, join can also be classified into join-all and join-some.

join-all In the join-all case, it is required that all the signatures are present and valid. In the fork-all example, the MIS department are required to collect all the signed copies from all the concerning departments, i.e. from the Financial Controller and the Estate Office. Furthermore, all these signatures have to be valid. Otherwise an exception on invalid signature should be raised.

join-some In the join-some case, there is no need to wait for all the signatures, but only those which are mandatory and those which satisfy the conditions. In the fork-some case, the MIS manager may go ahead with the renovation work after (s)he has received the valid signatures from the Financial Controller, Estate Office and any three BoD members. It should be noticed that the manager may have sent copies to more than three BoD members. Furthermore, this is a case of voting where only a certain number of signatures are sought.

3.3 Anonymous Signature

Anonymous signatures are often used in group decision making processes. This is a kind of multiple signature in which copies of the forms are sent to those actors concerned for signature. Special arrangements have to be made to hide the identity of the actors who sign the signatures. Additional mechanism is also required to ensure that the signature is signed by valid actors.

4 Realization of Multiple Signature in Workflow Systems

To handle multiple signatures in workflow systems, in addition to making use of the digital signature technique, we also need to incorporate some data structures and mechanisms into the workflow systems.

4.1 Signature On Signature

Many digital signature schemes are designed with only one signature in mind. There are two main approaches in handling multiple signatures. The first approach is to use complicate mathematical schemes, such as the modified RSA scheme [15] to handle multiple signatures. These schemes are usually very complicated in structure, and their application is not easily extendible to solve related problems. Therefore it is relatively difficult to incorporate them into existing workflow systems. The second approach is to apply single digital signatures more than once with some predefined sequences. This approach can be demonstrated by the following example.

Suppose three persons, Amy, Bob, and Charlie want to sign the same document, M.

1. Amy encrypts M with her private key. The encrypted message is denoted as SA. Amy sends M and SA to Bob.

2. After Bob receives M and SA, Bob decrypts SA with Amy’s public key to validate that SA is really generated by Amy’s private key. After that, Bob encrypts M and SA with his private key. The encrypted Message is denoted as SB. Bob sends M and SB to Charlie.
3. After Charlie receives M and SB, he decrypts SB with Bob’s public key to validate that SB is really generated by Bob’s private key. Then Charles decrypts SA with Amy’s public key to validate that SA is really generated by Amy’s private key.

This approach has the advantage that no extra mathematical schemes are used. All the cryptographic functions used are identical to those used in single signature schemes. The extra work is on how to combine different data items such as M, SA and SB in the above example. As a result, this approach is well-adapted for incorporating multiple signatures in an electronic workflow system.

Note that by migrating multiple signatures from a paper-based workflow system to an electronic workflow system, extra functionalities are created. In the above example, if carried out in a paper-based workflow system, there is no way to ensure that Bob signs the form after Amy has signed it. The appearance of the signatures on the paper form is identical, disregard the sequence of signing. However, in the electronic workflow system, we can ensure the sequence of signing. We call this mechanism signature on signature.

In the above example, the purpose of digital signatures is mainly for authentication of the owner of the documents. However, authentication is not the only function of a hand-written signature. Depending on the context of the paper-based message, hand-written signatures are also used for purposes, including but not limited to: authorization, generation of time stamp, providing evidences for accountability, and creating witness. These properties are related to the organization model and behaviour, and are usually ignored by research results in public key cryptography, which treats all types of content in a message as identical.

Therefore, applying public key cryptography to solve the signature problem of workflow is not sufficient [7]. Additional research effort is needed to study and incorporate the public key digital signature schemes into a workflow model, and this is the main theme of this paper.

4.2 Extensions in Liaison Workflow Model

In order to handle signatures for different purposes properly, we first extend the Liaison workflow model with appropriate facilities such that all the required information for handling different purposes of signatures can be captured and modelled in the workflow specifications.

Liaison is a comprehensive workflow model. Most of the information for handling different purposes of signatures, such as organisation structures, have been captured by the model. The only extension required by the Liaison workflow model is the extension in data types in Information Model.

In Liaison the Information Model section consists of two subsections: Data Model and Form Model. The Data Model section defines the fields and the data types in the forms to be used in the workflow system. The Form Model specifies the fields included in the forms and the presentation format for the forms.

The composite type signature consists of signature_content, purposes and associations. The signature_content and the purposes are mandatory fields. These two fields hold the digital signature and the purposes of the signature respectively. The purposes is a variable length field because a signature can used for more than one purpose. The associations field holds resources which may be required for the purposes of the signature. It is optional because not all purposes are associated with resources.

4.2.1 Data Model

In order to support multiple signatures, the composite data types owner, decision and anonymity have to be introduced to hold the necessary information.

owner The owner data type is used to hold the identity of the signer of a signature. This is used to provide the information of the signer for the validation of the signature. It is a mandatory field.

anonymity The anonymity data type holds the information whether the identities of the signers can be disclosed. The values of this type are Open and Anonymous.

decision The decision data type is a composite type which consists of at least one group. Each of the groups is also a composite type which holds the information on the number of signatures required from the specified group of signers. In the case of fork or join, It also contains the policy as to whether the task can proceed without waiting for all signatures. In terms of DeMarco’s data dictionary notation, the data type decision are expressed as follows.

\[
\begin{align*}
\text{decision} & = 1\{\text{group}\} \\
\text{group} & = \text{anonymity} + \text{number} + \\
& \quad \text{signerList} (\text{policy}) \\
\text{policy} & = [\text{All} | \text{Min}] \\
\text{number} & = \text{integer} \\
\text{signerList} & = 1\{\text{signer}\}
\end{align*}
\]

All signers in the signerList are those actors who may sign the form. They must have been registered in the Organisation Model. Furthermore, there is at least one
The design of different multiple signature mechanisms in workflow systems is discussed in the following section.

5 Design of Different Multiple Signature Mechanism

5.1 Sequential Multiple Signatures

In handling sequential multiple signatures, signing and validation of signatures usually go hand-in-hand. Before signing a signature, the Signature Manager has to validate whether the previous actors are eligible actors to sign the form. These include whether the actors who signed the form are:

- the right actors who are assigned to process the tasks and
- are eligible to process the tasks.

If any of the signatures are invalid, this means that the form has been processed by some invalid actors. Information may have disclosed to some inappropriate actors or some inappropriate processing might have been carried out. Hence exceptions should be raised.

In the case of dependent sequential multiple signatures, the sequence of the actors signing the form has to be checked. If the sequence of signing the form is incorrect, the signing processing should be rolled back to re-do the signing again.

According to the dependency requirement, signatures can be signed differently.

Independent Sequential Multiple Signature In the independent sequential multiple signature case, since the sequence of signing the signatures are immaterial, the signers can simple sign on the context of the form without signing on the signatures of the previous signers. All the signatures together with the owner are simply included in the form for later processing.

In the validation of this case, the system only has to check:

1. the validity of each of the signatures and
2. all the form contexts obtained from the digital signatures are consistent.

Dependent Sequential Multiple Signature In the case of dependent sequential multiple signatures, the sequence of signing the forms is important. This can be solved easily by applying the signature on signature mechanism. This means that the signers has to sign not only on the context of the form but also the signatures of the previous signers to form the new digital signature.

In the validation of the signature, from the owner, the identity of the previous signer can be obtained. Then the signature can be validated by decrypting the signature with the public key of the previous signer. After decrypting the signature, the form content and the signature of the “signer before this signer” will be obtained. Then the signature of this signer can be validated in a similar manner. This validation process is repeated until all the signatures are validated.

5.2 Parallel Multiple Signature

In parallel multiple signatures, some information about the potential signers of the forms has to be kept in the form. Otherwise, it will be difficult to get the right information to process with the signatures.

With the designed data types group in Section 4.2.1, it is easy to keep the information of:

- the number of potential signers who will receive the form;
- the potential signers who will sign on the form; and
- the number of signature required.

The potential signers who will receive the form for signatures are kept in the signerList. Then the number of potential signers who will receive the form can be obtained from this list. The number field holds the number of signatures required by the form. In our previous example, the signature of Financial Controller, Estate Office and 3 members of BoD are required in order to carry out renovation. This requirement can be denoted as

\[ (2, \{\text{Financial Controller, Estate Office}\}), \]
\[ (3, \{\text{Peter, Paul, Mary, Bob, Salina, Olivia}\}, \text{Min}) \]

In the first group, there are only two signers in the list and the number of signatures required is two. These two signatures are mandatory. In the second group, there are six signers in the list but the number of signatures required is only three. So any three of the signatures would be sufficient.

This example also gives the reason for making policy an optional field. If the group is a mandatory group, the processing has to wait for all signatures, so there is no need to record the policy.

Depending on the requirement of the workflow, if all the signatures in the signerList have to be collected before the
workflow can proceed, the value All should be put in the policy. Otherwise, the value Min should be put there.

This data structure design elegantly captures the requirements of fork-all and fork-some.

The signing process of parallel multiple signature is handled as a simple signing process because there are multiple copies of the same form and each signature is signed over the form individually.

In the case of join, with this data structure, both join-all and join-some can be handled easily.

**join-all** There are two possible cases for join-all. If the policy field is absent, all the signatures in the group have to be collected. If the value of policy is All, all the signatures in the group have also to be collected.

**join-some** If the value in the policy is Min, then it is the policy that the workflow can proceed when the minimum amount of signatures is obtained. There is no need to wait for all signatures in the signerList.

In validating the signatures for this purposes, the Signature Manager has to validate all the signatures over each of the form one by one. If there are any invalid signatures, the validation is regarded as invalid. Whether exceptions are raised when there are invalid signatures will depend on the purposes of the multiple signature. These purposes can be classified into mandatory and voting type.

**Mandatory** In the mandatory type, since all signatures are required in the workflow. Any invalid signature will invalid the process. Exceptions should then be raised for further action.

**Voting** In the voting process, each actor signs on their own copy of the form. The signing process is handled as a simple signature purpose. In the validation process, since invalid signatures will only invalid the votes, invalid signatures will not cause exceptions. Two counters have to be used for counting the valid and invalid votes. Furthermore, there should be a field in the form for the Signature Manager to indicate whether the form is a valid vote. The validated votes, together with the counting, are then returned to the task requesting signature validation.

### 5.3 Anonymous

The value of the anonymity field is anonymous means that the identities of the signers in the group cannot be disclosed. Anonymity is easier to be implemented in electronic workflow systems than in paper workflow systems. There are standard cryptographic techniques to implement voting schemes [15]. Signature Manager only needs to validate the signatures in the usual way. Afterwards, the Signature Manager only needs to send the validation result, which is either valid or invalid, to the corresponding tasks without providing any information about the actors who signed the form.

If the value of anonymity is open, this means that the identities of the signers are to be disclosed. No extra work is required.

### 6 Conclusion

This paper studied the problem of providing multiple signatures in an electronic workflow system. Multiple signatures are used for many different administrative reasons, as shown in Section 3. Although digital signature schemes that provide non-repudiation properties have been implemented successfully by public key cryptography, those schemes are insufficient to satisfy different purposes of signatures, especially multiple signatures, in workflow systems.

The contribution of this paper is firstly, to provide a classification of different administrative purposes of multiple signatures in a workflow system; and secondly, to propose a design of a signature manager in an electronic workflow system model which is able to handle multiple signatures; and finally, to give a design of how to make the signature manager being able to handle the different multiple signature purposes.

### References


